2Л

УДК 004.08

С.С. Литвин, К.А. Ручкин
Государственный университет информатики и искусственного интеллекта,
г. Донецк, Украина
s_lit@ukr.net, c_ruchkin@mail.ru

Разработка алгоритма распознавания замкнутых плоских кривых*

В данной работе продолжены начатые в [1-5] исследования по разработке алгоритма распознавания замкнутых плоских кривых (траекторий), построенных на двумерных сечениях Пуанкаре с помощью программы Modeler [1]. Предложенный алгоритм обработки изображений использует методику заливки и разработанный классификатор изображений. Алгоритм был реализован на языках высокого уровня С++ и Action Script 3.0 в среде Borland Builder С++ 6.0 и Macromedia Flash Animation соответственно, протестирован на более чем 200 тестовых экземплярах изображений и показал достаточную эффективность.

Введение

Как известно [1], решение задачи прогнозирования регулярного и хаотического поведения нелинейных динамических систем сводится к анализу вида некоторых графических изображений – сечений Пуанкаре. При этом сам процесс исследования можно разбить на несколько этапов. На первом этапе используется процесс дискретизации решения динамической системы с помощью методов численного интегрирования. На втором этапе проводится графическое и геометрическое моделирование полученных результатов. Если размерность системы не превышает трех, результат этого интегрирования удобно представить в графическом виде на экране компьютера в виде последовательности точек и образующих некоторую кривую (траекторию) в пространстве, характеризующую состояние системы в любой момент времени. На третьем этапе проводится анализ вида этой траектории и делается вывод о характере поведения системы. В случае, когда размерность исследуемой системы превышает три, то результат численного моделирования системы в графическом виде в трехмерном пространстве можно представить только в виде сечений и проекций. Одним из видов таких сечений пространственных кривых является двухмерное сечение Пуанкаре.

Постановка задачи

Исходными данными являются изображения сечений Пуанкаре на сфере Пуассона, полученные с помощью разработанной программы Modeler [1]. Эти сечения построены в трехмерном пространстве и представляют собой сферу, на которой изображены множества точек (облака точек). В некоторых (регулярных) случаях эти множества точек образовывают замкнутую трехмерную кривую – окружность, лежащую на поверхности сферы. Проецирование этой окружности на плоскость позволяет выде-

* Работа выполнена в рамках госбюджетной тематики «Разработка теоретических основ и средств создания информационных систем с использованием современных компьютерных технологий» (шифр ПОИС-20 09) ГУИ и ИИ.

лить четыре случая: линия, окружность, эллиптическая и овальная кривая (рис. 1). Разработке алгоритмического и программного обеспечения для распознавания этих случаев посвящены дальнейшие исследования.

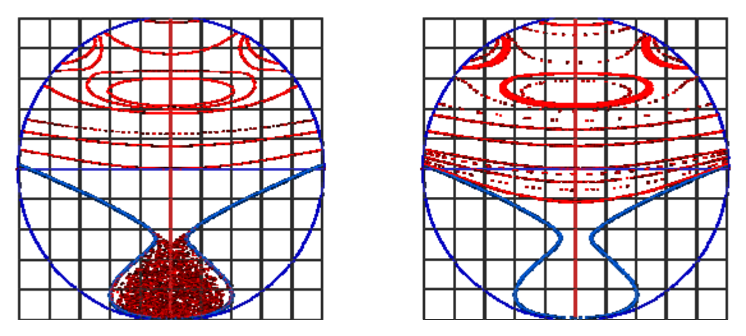


Рисунок 1 – Экземпляры распознаваемых изображений

Таким образом, **целью работы** является разработка методов и алгоритмов решения задачи распознавания замкнутых кривых, которые представляют собой траектории, построенные на двухмерных сечениях Пуанкаре.

Объектом исследования выступают методы кластеризации и алгоритмы распознавания плоских кривых на двухмерных изображениях.

Предметом исследования являются изображения специального вида с хаотическими и нехаотическими траекториями.

Алгоритм обработки и распознавания кривых

В основе алгоритма лежит предварительная, основная и дополнительная обработка изображения, которые проводятся в 6 этапов. Каждый этап получает на вход результат работы предыдущего шага и выдает результат для следующего. На рис. 2 представлена схема взаимодействия модулей.

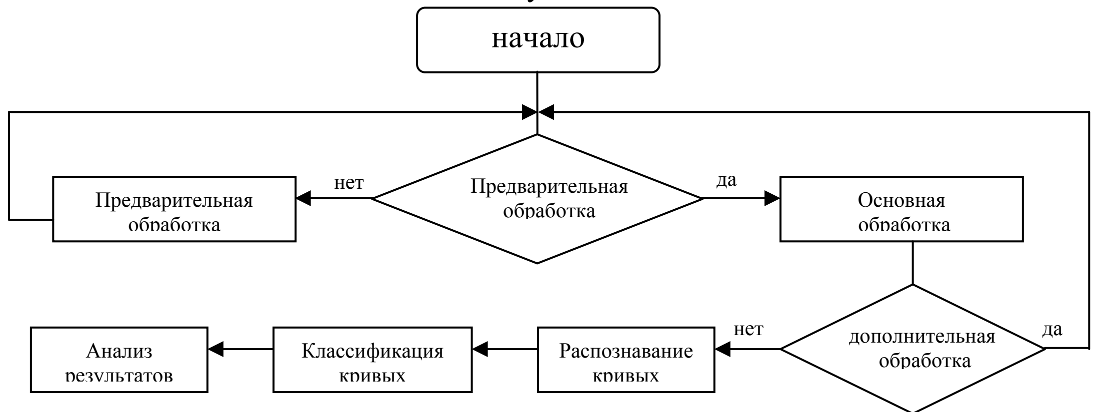


Рисунок 2 – Схема взаимодействия модулей

Опишем алгоритм более подробно.

Шаг 1. Предварительная обработка. Предварительная обработка необходима для устранения объектов на изображении, способных увеличить ошибку распознавания. Таковыми являются – вспомогательная сетка (фон), координатные оси, контуры объекта (опционально), одиночные пиксели, недостроенные траектории.

Очистка изображения от вспомогательных линий и фона. Очистка производится путем перебора всех пикселей входного изображения. При этом точки, имеющие черный цвет (#000000), заменяются на белый (#ffffff). Это необходимо для дальнейшего анализа траекторий, который будет использовать коды красного (#ff0000) и зеленого (#00ff00) цветов, описанного в шаге 4.

На изображении присутствует ось ординат (OY) красного цвета, являющаяся помехой для обнаружения кривых. Для глаза она имеет тот же оттенок, что и искомые кривые, но код её цвета не совпадает с ними. Ось располагается на заранее известных координатах (199×0 – 199×299, 200×0 – 200×299, при разрешении изображения 400×300 пикселей). Для корректного удаления оси нужно проверить соседние точки по горизонтали (координаты 198×0 – 198×299, 201×0 – 201×299) и произвести замену этими пикселями. Таким образом, удается избежать «рассечения» кривых. На рис. 3 представлен алгоритм предварительной обработки изображения.

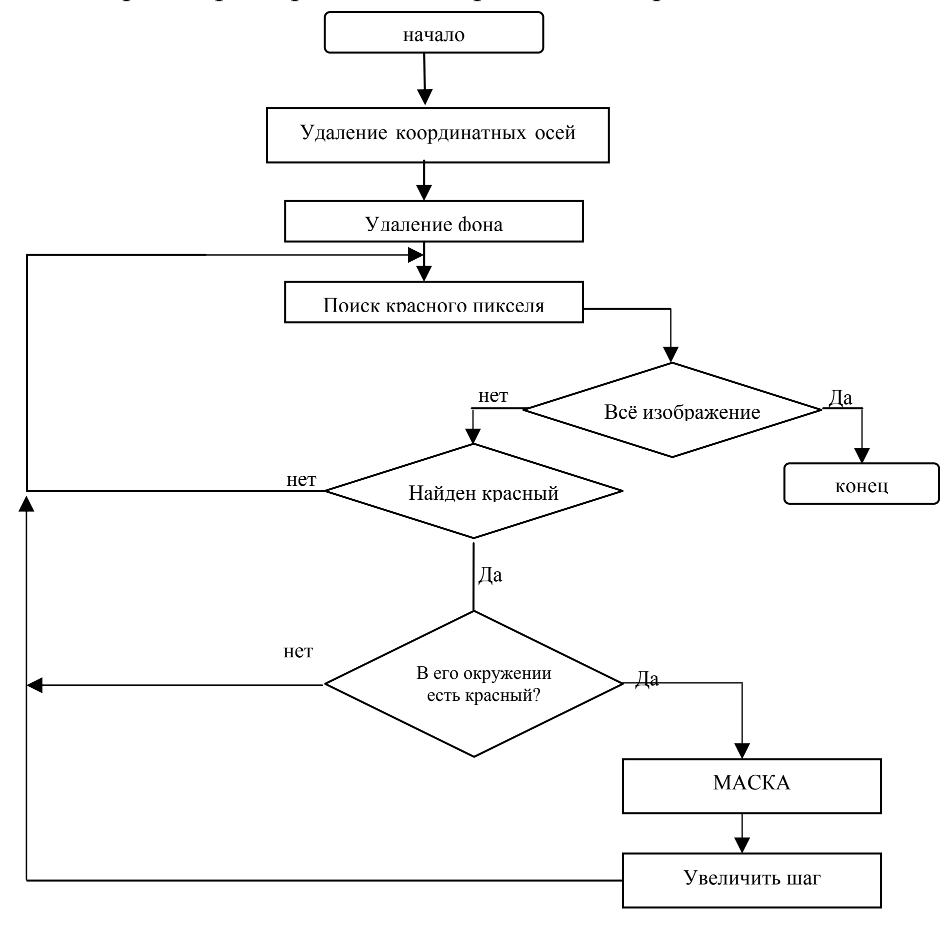


Рисунок 3 – Алгоритм предварительной обработки

Шаг 2. Основная обработка изображения. В процессе построения кривых программой-генератором возникают ситуации, при которых траектория является незавершенной. Эти случаи вызваны ограничением по времени, либо когда полное построение не представляется возможным. В виду того, что эта кривая строится по особому алгоритму, представляется возможным восстановление потерянных (недостроенных) сегментов траектории. Образцы траекторий с различной степенью завершенности представлены на рис. 4 – 6.

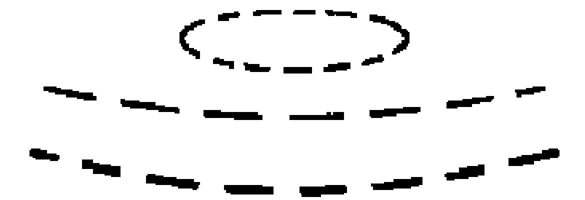


Рисунок 4 – Траектории завершены на 50%

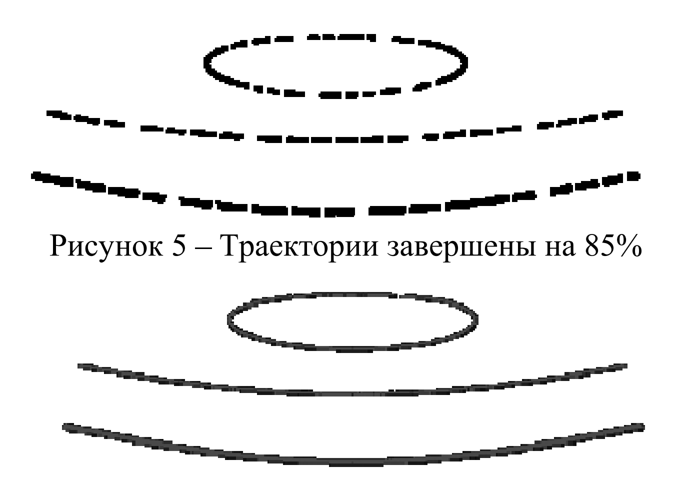


Рисунок 5 – Траектории завершены на 85%

Рисунок 6 – Траектории завершены на 100%

Алгоритм восстановления отсутствующих сегментов траектории основан на цветовом анализе изображения. Был применен пороговый метод – если яркость данного элемента превышает среднюю яркость локальной окрестности, тогда яркость данного элемента заменяется на среднюю яркость окрестности. Если полученная яркость превышает заданный порог, тогда точка принимает значение цвета кривой (#ff0000), в противном случае – белый (#ffffff).

Алгоритм эффективен в случаях, при которых потери сегментов составляют не более 20%. В остальных случаях появляется риск замены пустых областей внутри замкнутых траекторий на цвет контура (случаи, когда ширина замкнутой траектории в 2 и более раз меньше длины).

В зависимости от степени сложности (в это понятие входит процент потери сегментов на кривой, соотношение ширины к длине траектории, разброса и других помех) обрабатываемого изображения возможна повторная обработка.

Шаг 3. Алгоритм распознавания кривых. Выделение областей изображения с известными свойствами (признаками) кривых.

На входе алгоритма имеется подготовленное изображение с объектами (#ff0000), очищенное от шумов, вспомогательной сетки и границ объекта. Так как исследуемые кривые были получены с помощью сечения трехмерной модели сферического тела, они обладают свойством симметричности по вертикали и располагаются строго симметрично относительно центра изображения.

Исходя из этого, оптимальной точкой для начала поиска кривых является координата 199×299 (для разрешения изображения 400×300). Направление поиска осуществляется по формуле 199×N (где N = 299…0) снизу вверх. Эта нижняя центральная позиция увеличивает вероятность обнаружения кривых, т.к. они обязательно пройдут через середину. Поиск осуществляется до тех пор, пока не выполняется одно из условий – найден контур объекта, либо достигнута граница изображения.

После того как был найден контур объекта, происходит обход вдоль контура по принципу «правой руки». Во время обхода контура оставляется «след» зелёного цвета (#00ff00), нужный для остановки в случае движения по замкнутому контуру, а также визуального выделения области. Обход прекращается по достижении одного из условий – следующий пиксель имеет зелёный цвет (#00ff00), либо достигнута граница сферы (на плоскости является окружностью).

Контур считается выделенным. Продолжается поиск следующей траектории по тому же принципу, но уже начиная с координаты 199×K (где K – значение позиции первого зеленого пикселя сверху по координате Y).

Результат работы алгоритма, примененного относительно кривой, завершенной на 85%, приведен на рис. 7.



Рисунок 7 – Траектории завершены алгоритмом

Шаг 5. Классификация кривых. Поиск замкнутых, незамкнутых кривых и хаотичных областей алгоритмом заливки.

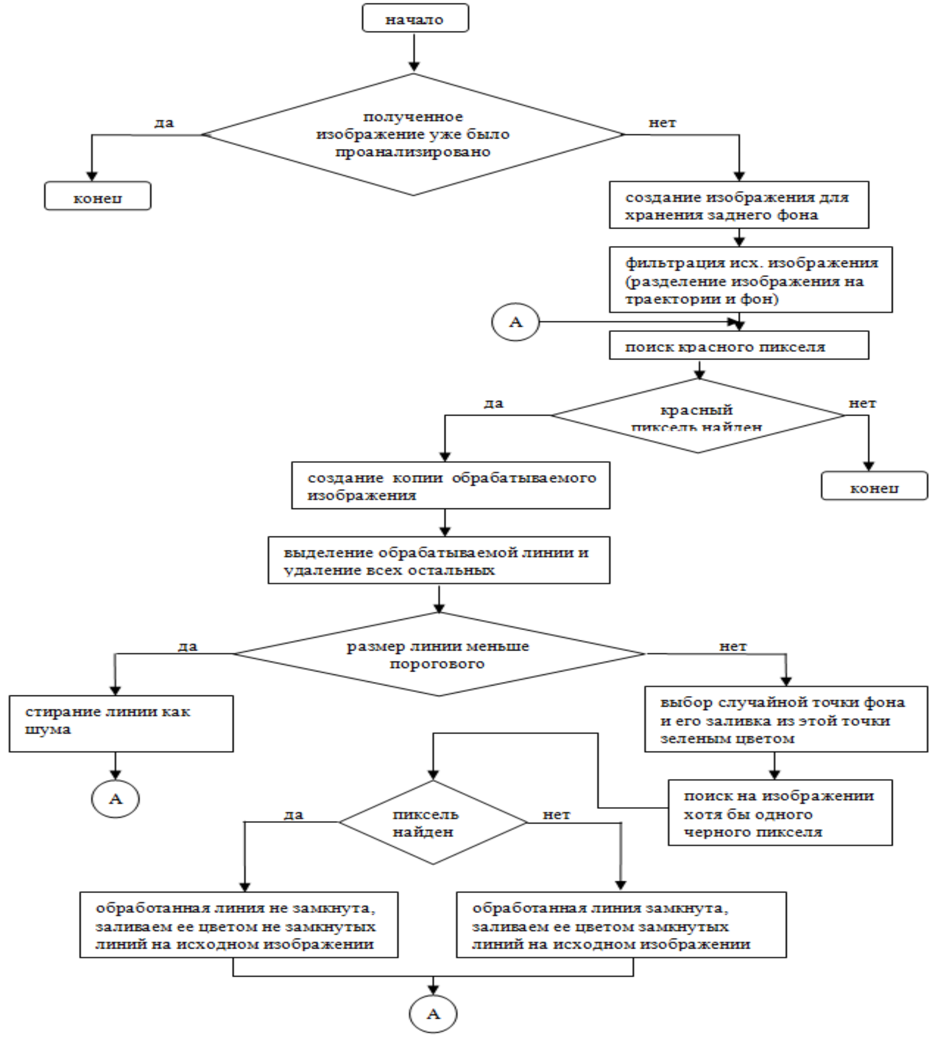


Рисунок 8 – Структурная схема работы алгоритма

После выделения границ контуров происходит поиск траекторий, который заключается в последовательном переборе всех пикселей изображения до нахождения пикселя красного цвета. Если пиксель был найден – создается временная копия изображения. На временной копии производится заливка найденной траектории синим цветом (#0000FF), с последующим отсечением всех остальных траекторий по цветовому признаку.

После отсечения всех траекторий, кроме одной, производится определение ее размера с последующей классификацией. Если линия слишком мала, то есть ее размер меньше порогового значения заданного, то она не подвергается дальнейшему анализу и считается шумом. В противном случае выбирается один из пикселей фона (цвет фона – #FFFFFF) и от него на временном изображении производится заливка зеленым цветом (#00FF00). После заливки производится поиск пикселей белого цвета, наличие которых свидетельствует о замкнутости траектории, а отсутствие о том, что она разомкнута. Данный алгоритм адекватно работает лишь после подготовки изображения на предыдущем этапе. На рис. 8 приведена структурная схема работы алгоритма.

Шаг 6. На последнем этапе проводится дальнейший анализ изображения. Анализ степени хаотичности распознанных кривых основывается на статистических данных обработанного изображения. Анализируется количество замкнутых и незамкнутых траекторий, неправильно распознанных или несуществующих областей.

Результаты тестовых испытаний

Для тестовых испытаний были отобраны однотипные образцы траекторий. Цель испытаний – выявление случаев неэффективного либо неадекватного поиска и выделения кривых. Образцы представляли собой как завершенные, так и незавершенные объекты. Наибольшая эффективность алгоритма проявилась на изображениях с отсутствием хаотичных областей и завершенностью кривых на 85 – 100%. Ошибочное распознавание хаотичных областей проявлялось в случаях с плотностью расположения точек на расстоянии меньшем либо равным размеру самой точки. Результаты работы алгоритма распознавания приведены на рис. 9 – 11.

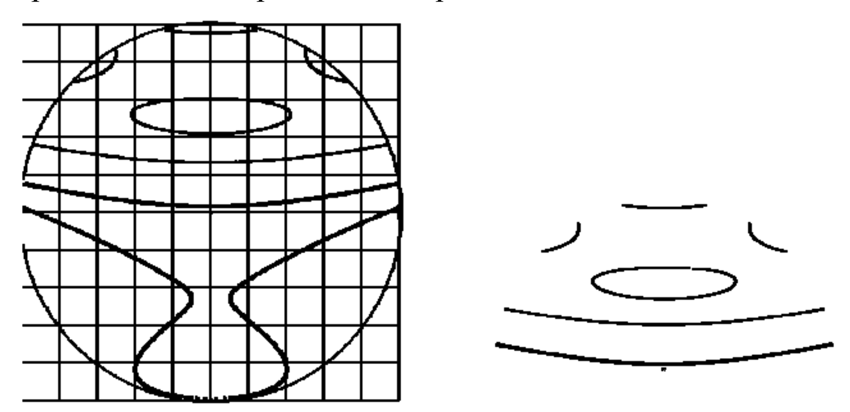


Рисунок 9 – Эффективное распознавание при идеальных условиях

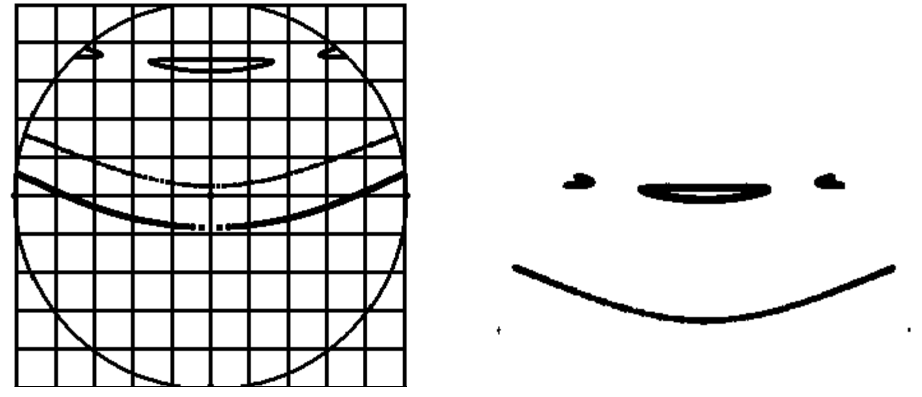


Рисунок 10 – Распознавание недостроенных траекторий



Рисунок 11 – Хаотическая область

В ходе проведенного тестирования программа показала не плохую скорость работ, на картинках разрешением 400×300 пикселей анализ производился в течение 1 – 3 с. Распознавание велось на картинках, смоделированных программой MODELER, при моделировании изображений были использованы параметры программы A = 1,70, B = 0,9, C = 0,8, h = 1,91, g = 1, nu1 = –0,8, nu2 = 0,6, nu3 = 0, mx = 1, my = 0, mz = 0. Выбор этих значений обусловлен тем, что при данных настройках возможно сгенерировать любой необходимый вариант траекторий для демонстрации возможностей ПО.

Расчет сложности алгоритма

Для оценки производительности алгоритма, его временной сложности, рационально использовать асимптотический подход, т.к. алгоритм распознавания включает в себя более мелкие подгруппы алгоритмов и операций, зависящих от одних и тех же параметров и дающих несущественный прирост сложности при значительном увеличении объема входных данных.

За единицу времени принимается группа операций считывания одного пикселя. Характер алгоритма таков, что при увеличении количества траекторий на изображении количество анализируемых кластеров увеличивается, следовательно, увеличивается количество операций считывания/записи (заливка контуров). Характер сложности в данном случае носит следующую зависимость, которая представлена на рис. 12.

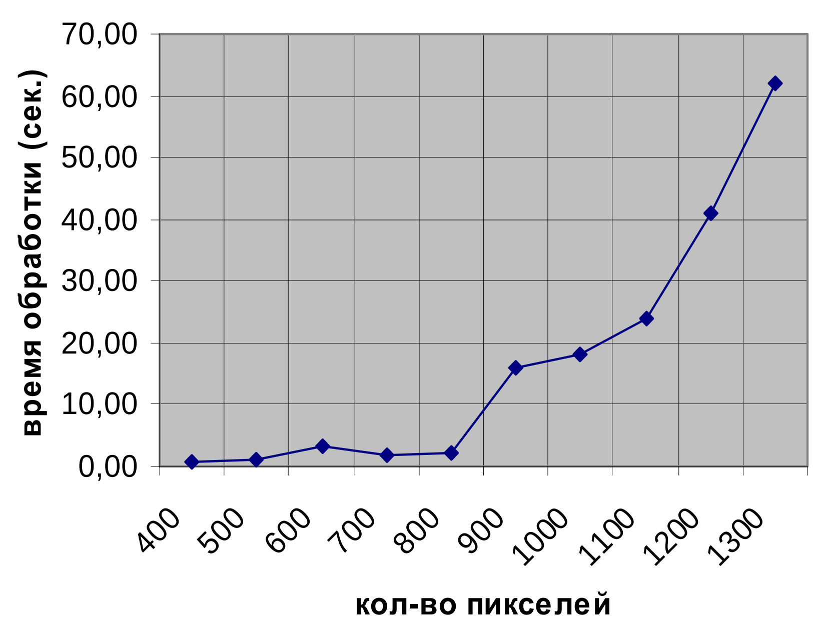


Рисунок 12 – График зависимости времени работы алгоритма от размера входных данных

Выводы

В данной работе рассмотрены основные способы распознавания образов (изображений), а также их программная реализация. В ходе работы была произведена постановка задачи распознавания замкнутых и не замкнутых траекторий на плоскости. В частности, был проведен анализ существующих методов распознавания изображений, исследованы принципы и спецификации хаотических и нехаотических кривых на специального вида изображениях, были выделены основные признаки для построения классификатора, разработано алгоритмическое обеспечение для распознавания кривых внутри класса, были проведены тестовые испытания и анализ результатов, произведен анализ эффективности алгоритма.

Литература

1. Ручкин К.А. Методы компьютерного моделирования и анализа решений задач хаотической динамики / К.А. Ручкин // Искусственный интеллект. – 2004. – № 4. – С. 175-181.
2. Ручкин К.А. Разработка компьютерной системы для построения и анализа сечений Пуанкаре / К.А. Ручкин // Искусственный интеллект. – 2009. – № 1. – С. 83-87.
3. Гашененко И.Н. Визуальное моделирование хаотической динамики тяжелого твердого тела / Гашененко И.Н., Лапенко С.В., Ручкин К.А. – Укр. матем. конгр., 2001. – С. 14-15.
4. Малинецкий Г.Г. Хаос. Структуры. Вычислительный эксперимент. Введение в нелинейную динамику / Малинецкий Г.Г. – М. : УРСС, 2001. – 240 с.
5. Форсайт Д. Компьютерное зрение. Современный поход / Форсайт Д. – М. : Вильямс, 2004. – 610 с.

С.С. Литвин, К.А. Ручкін

Розробка алгоритму розпізнавання замкнених плоских кривих
У поданій роботі продовжені розпочаті у [1-5] дослідження щодо розробки алгоритму розпізнавання замкнених плоских кривих (траєкторій), побудованих на двовимірних перерізах Пуанкаре при допомозі програми Modeler [1]. Запропонований алгоритм обробки зображень використовує методику заливки та розроблений класифікатор зображень. Алгоритм був реалізований мовами високого рівня С++ та Action Script 3.0 у середовищі Borland Builder C++ 6.0 та Macromedia Flash Animation відповідно, протестований на більш ніж 200 екземплярах для тестування зображень і показав достатню ефективність.

S.S. Litvin, K.A. Ruchkin

Development of Algorithm of Recognition of the Closed Plane Curves
The present work continues researches begun in [1-5] on development of algorithm of recognition of the closed plane curves (trajectories) constructed on two-dimensional cuts of Poincare by means of program Modeler [1]. The offered algorithm of image processing uses a technique of pouring and the developed qualifier of images. The algorithm has been realised in languages of high level C ++ and Action Script 3.0 in the environment Borland Builder C ++ 6.0 and Macromedia Flash Animation accordingly, tested on more than 200 test copies of images and has shown sufficient efficiency.

Статья поступила в редакцию 25.10.2010.